

Plano de Ensino

1) Identificação

Disciplina: INE5408 - Estruturas de Dados
Turma(s): 03208A, 03208B
Carga horária: 108 horas-aula Teóricas: 54 Práticas: 54
Período: 1º semestre de 2020

2) Cursos

- Ciências da Computação (208)

3) Requisitos

- Ciências da Computação (208)
 - INE5404 - Programação Orientada a Objetos II

4) Professores

- Aldo Von Wangenheim (aldo.vw@ufsc.br)
- Alexandre Goncalves Silva (alexandre.goncalves.silva@ufsc.br)

5) Ementa

Alocação dinâmica de memória. Variáveis estáticas e dinâmicas. Estruturas lineares. Tabelas de Espalhamento. Árvores. Árvores de Pesquisa. Métodos de ordenação. Métodos de acesso a arquivos. Técnicas de implementações iterativas e recursivas de estruturas de dados. Complexidade dos algoritmos em estruturas de dados.

6) Objetivos

Geral: Capacitar o estudante a compreender, tanto do ponto de vista conceitual e teórico quanto prático (implementação e solução de problemas reais), as estruturas de dados clássicas a partir da perspectiva de diferentes paradigmas de programação.

Específicos:

- Identificar o papel das estruturas de dados no desenvolvimento de software.
- Compreender a teoria associada a cada modelo, sendo capaz de compreender e aplicar adequadamente aspectos de complexidade de algoritmos associados.
- Capacitar o aluno a criar uma biblioteca de estruturas de dados reutilizáveis.
- Identificar as estruturas de dados pertinentes a um problema dado.
- Transferir seus conhecimentos a problemas práticos do mundo real, sendo capaz de solucionar estes problemas práticos do mundo através da utilização das Estruturas de Dados adequadas, tanto do ponto de vista da eficácia quanto da eficiência.

7) Conteúdo Programático

- 7.1) Alocação dinâmica de memória [6 horas-aula]
 - Variáveis estáticas e dinâmicas
 - Ponteiros
 - Passagem de parâmetros por referência, valor e nome
- 7.2) Estruturas lineares [24 horas-aula]
 - Listas
 - Pilhas
 - Filas
- 7.3) Complexidade de algoritmos [6 horas-aula]
 - Conceitos de Complexidade de Algoritmos
 - Métodos práticos para Análise da Complexidade de Estruturas e Algoritmos
- 7.4) Árvores [24 horas-aula]
 - Árvore binária

- Árvores binárias semibalanceadas (Árvore AVL, Árvore Red-Black)
 - Árvore semibalanceadas multivias (Árvore B e B+)
 - Árvores binárias multichaves (Árvore k-d)
 - Árvores semibalanceadas multivias multichaves (Árvores k-b-d e b-k-d)
- 7.5) Tabela de espalhamento (hash) [15 horas-aula]
- Tratamento de colisões
 - Funções de espalhamento
- 7.6) Métodos de ordenação [12 horas-aula]
- Conceitos básicos, implicações e premissas
 - Métodos de Complexidade Quadrática
 - Método por inserção
 - Método por seleção
 - Método da bolha
 - Métodos Avançados e de Complexidade $n \log n$
 - Método Quicksort
 - Método Heapsort
- 7.7) Estruturas de dados em arquivo [21 horas-aula]
- Acesso direto e Acesso sequencial
 - Organização de Arquivos de Índices
 - Métodos indexados seqüenciais, ISAM e VSAM
 - Indexação por Árvore
 - Indexação por Multilistas e Lista Invertida

8) Metodologia

A disciplina terá um enfoque eminentemente prático. A metodologia de ensino será baseada no contraponto entre Aulas Teóricas e Aulas Práticas. Para tanto todo novo assunto será introduzido em uma aula teórica. Este conteúdo teórico será fixado por meio de atividades de caráter prático, podendo ser executadas com auxílio de um Estagiário de Docência ou Monitor, onde serão realizadas modelagem e implementação com o objetivo de fixar o conteúdo, além da discussão em grupo de problemas de compreensão e implementação encontrados pela turma. Como linguagem de programação para o ensino do conteúdo fica determinada a Linguagem C++ e como ambiente de implementação obrigatoriamente a Plataforma Linux/Unix, sendo a ferramenta de desenvolvimento preferencial o ambiente VPL do Moodle, podendo outras ferramentas serem utilizadas. Haverá uma breve introdução a ferramentas de produtividade nessas plataformas, à medida que tornarem necessárias, principalmente as de teste unitário, mas fica a cargo do aluno a responsabilidade de adquirir proficiência nas mesmas. O ensino dos conteúdos se dará por meio da utilização progressiva do Paradigma Orientado a Objetos, de acordo com a adequação à melhor compreensão do conteúdo. A ferramenta de EAD Moodle, disponível em moodle.ufsc.br, será utilizada para guiar e organizar o ensino, sendo o repositório oficial de material de aula, devendo os alunos constantemente consultá-la para saber de novos exercícios e trabalhos requeridos. A disciplina no Moodle também detalhará o cronograma deste plano de ensino, servindo para marcar os prazos das atividades avaliativas e documentar alterações de cronograma advindas de necessidades identificadas no semestre. O fórum de discussão no ambiente do Moodle será utilizado para intermediar a comunicação entre professor, estagiário de docência, monitor e alunos. Para efeitos da avaliação da participação do aluno na disciplina, as suas estatísticas de utilização da ferramenta de EAD poderão ser levadas em consideração. Como recurso didático auxiliar o professor poderá empregar vivências na forma de jogos de aprendizado (*serious games*), a serem jogados em equipe ou na forma de objetos de aprendizado digitais, disponibilizados via Moodle, para uso em casa. A ferramenta Moodle ainda poderá ser utilizada para a entrega e avaliação automatizada, baseada em testes programáticos dos trabalhos de programação e de tarefas assinaladas pelo professor.

Pressupostos da metodologia de ensino: A metodologia adotada nesta disciplina pressupõe que os alunos do curso não se limitem a assistir às aulas, mas que utilizem para as atividades extras associadas a esta disciplina (leituras, resolução de exercícios teóricos e exercícios com o uso da ferramenta de EAD empregada). Pressupõe-se que os alunos tenham estudado o conteúdo utilizando a bibliografia indicada e tenham resolvido, como atividade extra, todos os exercícios propostos pelo professor. Atividades experimentais serão acompanhadas pelo aluno de Pós-Graduação do PPGCC, Allan Cerentini, que estará realizando seu estágio de docência em 2020-1.

9) Avaliação

A avaliação regular dos alunos se dará por meio de um conjunto de oito notas, sendo três destas notas referentes a trabalhos práticos, duas notas referentes a provas teóricas, duas notas referentes a provas práticas de programação e uma nota referente a testes de conhecimento. A média final será composta pela média ponderada destas 8 notas conforme explanado abaixo:

- **N1 (peso 2, assíncrono, entrega pelo Moodle)** - Média simples de notas atribuídas a pequenos trabalhos de implementação, denominados **Testes Práticos**, para a prática de programação. Estes trabalhos estarão agendados no Moodle e serão em número entre 10 e 20, de acordo com a avaliação do professor do perfil de aprendizado da turma e suas dificuldades.

- **N2 (peso 1, assíncrono, entrega pelo Moodle)** - Um projeto de implementação, denominado **Projeto I**, envolvendo um problema prático do mundo real solúvel com a aplicação de pilhas, listas ou filas e ponteiros e alocação dinâmica de memória. Este trabalho poderá ser realizado em equipe e poderá ser defendido pela equipe em casos onde não fique claro a participação de todos os membros da equipe, sendo atribuída nota individual a cada aluno.
- **N3 (peso 1, preferencialmente síncrono, pelo Moodle)** - Uma prova de prática de programação individual, denominada **Prova Prática I**, cobrindo os assuntos de estruturas lineares. A prova prática será feita utilizando a ferramenta Moodle na qual uma estrutura de dados deverá ser implementada e deverá passar por testes unitários automatizados de acordo com o preceitos utilizados em competições de olimpíadas de programação. A prova pode incluir restrições de tempo para implementação, tempo para execução e quantidade e técnicas de gerência de memória para execução.
- **N4 (peso 1, preferencialmente síncrono, pelo Moodle)** - Um prova teórica individual, denominada **Prova Teórica I**, cobrindo os assuntos de estruturas lineares e complexidade de algoritmos, realizado por meio do uso do Moodle, onde serão avaliados aspectos do conhecimento teórico.
- **N5 (peso 2, assíncrono, entrega pelo Moodle)** - Um Projeto de Implementação, denominado **Projeto II**, envolvendo um problema prático do mundo real solúvel com a aplicação de Técnicas de Gerência de Arquivos Indexados. Este trabalho poderá ser realizado em equipe e poderá ser defendido pela equipe em casos onde não fique claro a participação de todos os membros da equipe, sendo atribuída nota individual a cada aluno.
- **N6 (peso 1, preferencialmente síncrono, pelo Moodle)** - Uma prova de prática de programação individual, denominada **Prova Prática II**. A prova prática será feita utilizando a ferramenta Moodle na qual uma estrutura de dados deverá ser implementada e deverá passar por testes unitários automatizados de acordo com o preceitos utilizados em competições de olimpíadas de programação. A prova pode incluir restrições de tempo para implementação, tempo para execução e quantidade e técnicas de gerência de memória para execução.
- **N7 (peso 1, preferencialmente síncrono, pelo Moodle)** - Um prova teórica individual, denominada **Prova Teórica II**, cobrindo todo o conteúdo da disciplina, realizado por meio do uso do Moodle, onde serão avaliados aspectos do conhecimento teórico.
- **N8 (peso 1, assíncrono, pelo Moodle)** - Média simples de notas atribuídas a teste objetivos rápidos, denominados **Testes Teóricos**, a cada aproximadamente 5 itens de ementa, para reforço de aprendizagem teórica.

Portanto, a média final é dada por:

$$\cdot \text{MF} = (2N1 + N2 + N3 + N4 + 2N5 + N6 + N7 + N8) / 10$$

Todos os trabalhos deverão ser entregues juntamente com a documentação exigida pelo professor, sendo a avaliação realizada sobre código e documentação. A forma de entrega é até a data determinada e por meio do Moodle de acordo com os critérios estabelecidos para cada trabalho, podendo ser em arquivos compactados ou por meio da ferramenta de avaliação automática VPL, dependendo da indicação no enunciado do trabalho. É de responsabilidade do aluno entregar o trabalho na forma correta, arquivos corrompidos ou ilegíveis não serão considerados. Trabalhos entregues após o prazo sofrerão penalizações de 50% da nota independente do atraso. A critério do Professor, sendo o desempenho parcial dos alunos considerado adequado e suficiente, o que será mensurado pela qualidade da maioria dos trabalhos entregues (condição mínima: média da turma ≥ 7.0), o item de avaliação número quatro poderá ser considerado desnecessário e omitido. Também a critério do professor, sendo o desempenho parcial dos alunos considerado adequado e suficiente, o que será mensurado pela qualidade da maioria dos trabalhos entregues (condição mínima: média da turma ≥ 7.0), o item de avaliação número sete poderá ser considerado desnecessário e omitido. Em função do caráter prático da disciplina, os itens 1, 2, 3, 5 e 6 sempre serão avaliados. Nesse caso, a média final transferirá o peso das provas teóricas para as provas práticas correspondentes. Se o desempenho médio dos alunos durante o semestre for considerado extraordinário (condições mínimas: média geral da turma ≥ 7.0 ou média da turma no item de avaliação número três ≥ 7.0), o professor pode optar por não exigir a defesa dos Projetos de Implementação, considerando, daquelas equipes que demonstraram bom aproveitamento em todos os aspectos, apenas código e documentação entregues para fins de avaliação. Durante a defesa dos projetos de implementação, o professor se reserva o direito de questionar individualmente os alunos da equipe sobre aspectos teóricos da disciplina contemplados no trabalho, sendo o resultado desses questionamentos levado em consideração de forma individual na atribuição do conceito. Para avaliação dos trabalhos práticos o professor poderá trabalhar com dois modelos de correção: um manual e um automático, sendo dada preferência a este último sempre que possível. Para os casos de correção automática dos trabalhos de programação, estes serão avaliados por meio da plataforma Moodle. Os testes estarão disponíveis no momento da entrega do trabalho pelos alunos. Os testes serão aplicados e uma nota final do trabalho será sugerida pela ferramenta de correção, podendo o aluno entregar o trabalho quantas vezes forem necessárias para a obtenção da nota desejada, tendo como limite a data de entrega final do trabalho em questão. Cabe ao professor a confirmação da notas em avaliação posterior. Neste caso a nota pode ser alterada levando em consideração a técnica, o estilo e a eficiência do código à critério do professor e previamente estabelecido no enunciado do trabalho. Os trabalhos corrigidos automaticamente devem passar na sua integralidade e sem quaisquer defeitos nos testes programados para aquela entrega. Para tal os alunos devem sempre testar seus trabalho na plataforma de avaliação, não sendo aceitas reclamações pela não execução no ambiente

disponibilizado. Os testes aplicados no processo de avaliação automática podem incluir:

- **Testes baseados na interface de usuários**, onde dados são alimentados para o programa desenvolvido por meio da sua interface e o retorno é avaliado. Nestes casos a precisa implementação da interface é parte da avaliação.
- **Testes unitários em classes**, onde os alunos serão instruídos a implementar uma interface de classe para que as classes entregues possam ser testadas método a método quanto ao seu funcionamento.
- **Avaliação estática de código**, buscando boas práticas de nomeação de atributos, de visibilidade dos métodos, de estruturação e documentação do código.
- **Avaliação dinâmica de código**, buscando boas práticas de gerência de memória e consistência no uso de métodos e classes em tempo de execução.
- **Avaliação de desempenho**, utilizando restrições de tempo e memória para a execução dos trabalhos no sistema de avaliação disponibilizado pela plataforma Moodle.

Os pesos de cada item de avaliação nos trabalhos corrigidos automaticamente será divulgado junto com o enunciado de cada trabalho e podem variar de acordo com o objetivo didático de cada trabalho. No caso de correções manuais de trabalhos que não envolvam a defesa, serão utilizados 5 critérios objetivos, sendo que cada critério vale 2 pontos:

- **Compreensão do Problema:** entendeu o que era para fazer (1 ponto) e modelou corretamente a solução (1 ponto) ?
- **Emprego dos Algoritmos e Estruturas:** empregou os algoritmos e estruturas corretos (% dos algoritmos/estruturas necessários * 2 pontos) ?
- **Implementação dos Algoritmos e Estruturas:** os implementou corretamente (% dos algoritmos e estruturas necessários * 2 pontos) ?
- **Código:** compilou (1 ponto) e funcionou (1 ponto) ?
- **Documentação:** código está documentado e programado como indicado no início do semestre (comentários 1 ponto, estilo de codificação 1 ponto)?

Havendo defesa dos Projetos de Implementação, a mesma é considerada Prova e o não comparecimento injustificado implica em conceito nulo nestes trabalhos, sendo os critérios de avaliação da defesa os abaixo:

- **Compreensão do Problema:** entendeu o que era para fazer ? (2 pontos)
- **Solução:** soube encontrar uma solução ? (2 pontos)
- **Conhecimento teórico:** compreendeu as implicações teóricas da solução escolhida ? (2 pontos)
- **Algoritmos:** possui compreensão dos algoritmos empregados e sabe descrevê-los ? (2 pontos)
- **Código:** compreende a implementação, sabendo detalhar aspectos de seu funcionamento ou de falhas que estão ocorrendo ? (2 pontos)

Para efeitos de defesa, os trabalhos práticos dos alunos deverão estar disponíveis nos links de entrega disponibilizados no Moodle da disciplina, não podendo ser trocados ou atualizados quaisquer arquivos. Todas as defesas poderão ser realizadas na presença de uma testemunha (preferencialmente o estagiário de docência, aluno de pós-graduação ou professor do INE) para protocolar as perguntas e respostas. **Plágio e Deslealdade Acadêmica:** Todas as atividades da disciplina seguirão um código ético severo, onde os trabalhos serão avaliados quanto a plágio de forma automatizada por uma ferramenta que o professor dispõe. Caso esta ferramenta indique a possibilidade de plágio, o professor conferirá automaticamente a nota ZERO a todos os envolvidos e reportará ao órgão competente da universidade para a abertura de processo disciplinar e a aplicação das punições regimentais previstas. Quanto aos casos de deslealdade acadêmica em provas, o professor lavrará um termo que será entregue ao órgão competente da universidade para a abertura de processo disciplinar e a aplicação das punições regimentais previstas. **Frequência na Disciplina:** Para a avaliação da frequência na Disciplina fica rigidamente expresso o entendimento disposto na resolução 017/Cun/97 que determina a necessidade de 75% de presença para os cursos presenciais da Universidade. Todos os registros de frequência serão feitos pela ferramenta Moodle e estarão disponíveis para consulta por parte dos alunos.

Objetivando atender à resolução de 21 de julho de 2020 que dispõe sobre o redimensionamento de atividades acadêmicas da UFSC, suspensas excepcionalmente em função do isolamento social vinculado à pandemia de COVID-19, e sobre o Calendário Suplementar Excepcional referente ao primeiro semestre de 2020, indicamos que as aulas sofrerão as seguintes adaptações:

- **Avaliações:** as provas práticas, antes feitas em laboratório sem acesso a internet (por meio do Moodle Provas), serão ainda feitas em tempo de aula, mas agora de maneira livre e remota. Casos excepcionais de realização dessas atividades síncronas avaliativas deverão ser reportados pelo aluno ao professor, de modo que se estabeleça um novo período para sua aplicação caso a caso.
- **Bibliografia online:** as nossas aulas são baseadas quase que exclusivamente em material multimídia produzido pelo próprio professor e tradicionalmente disponibilizado online. Capítulos de livros que oferecem textos complementares ao material produzido pelo professor também têm tradicionalmente sido disponibilizados online através do AVEA Moodle Presencial. Continuaremos dessa forma. A bibliografia adicional que indicamos no Plano de Ensino é composta por livros clássicos, na sua maioria disponíveis online em sebos virtuais a baixo custo, caso o aluno os deseje adquirir, e é indicada apenas em caráter complementar. Artigos científicos, cuja leitura porventura venhamos a indicar, estão disponíveis em sítios de preprints como arXiv ou ResearchGate ou então já são disponibilizados online pelo professor através do AVEA Moodle.

- **Contraponto entre aulas assíncronas/síncronas:** as aulas teóricas ocorrerão preferencialmente em modelo assíncrono, por meio de vídeos (que poderão ser baseados em aulas presenciais gravadas anteriormente e disponibilizadas no canal da disciplina ou em aula gravada, no formato de estúdio, especialmente para este fim). Nos horários alocados de aula, serão disponibilizadas salas virtuais síncronas, por plataforma de videoconferência, destinadas principalmente para revisão de conteúdos pontuais, esclarecimento de dúvidas e auxílio à resolução dos exercícios. Em situações onde não existir um vídeo com o assunto do dia, a íntegra da aula poderá ser realizada de forma síncrona. Neste caso, o professor disponibilizará esta aula no canal da disciplina para que todos possam reassisti-la e consultá-la posteriormente.
- **Plataforma para aulas síncronas/videoconferência:** usaremos a Conferência Web RNP, instalada pelo SETIC/UFSC, ou a plataforma Google Meet, objetivando inclusive a participação das aulas pelo telefone celular. As salas virtuais criadas serão usadas para as aulas síncronas e para discussões de assuntos da disciplina ao longo de todo o semestre.
- **Plataforma para videoaulas assíncronas:** Serão utilizados o tradicional canal da disciplina na plataforma YouTube ou hospedagem em servidores da UFSC. Todos os links serão indicados no Moodle da disciplina contendo a íntegra das aulas teóricas.
- **Plataforma para discussões assíncronas:** para este fim está mantido o Fórum da Disciplina no Moodle, que temos tradicionalmente utilizado com sucesso. Havendo necessidade de uma discussão mais dinâmica poderá ser utilizada a plataforma Google Hangouts. Um link para o grupo de Whatsapp da disciplina também foi disponibilizado.
- **Registro de frequência:** o ensino se dará no modelo remoto e a frequência continua sendo obrigatória. Para registro da frequência de forma simplificada vamos utilizar o método da autoindicação de presença via Moodle Presencial pelo aluno. Para tanto o módulo de autoindicação de presença via Moodle pelo aluno será aberto no início de cada aula. De modo alternativo, os registros de participação nas sessões síncronas também contabilizarão presença nos horários de aula.
- **Aulas práticas:** as aulas práticas serão realizadas na forma de Estudo Dirigido, acompanhado de forma síncrona online pelo professor e pelo Estagiário de Docência da disciplina. Adicionalmente o acompanhamento também ocorrerá de forma assíncrona, seguindo a metodologia já definida mais acima no Plano de Ensino.
- **Direito de uso de imagem:** fica preservado o direito de imagem do professor e dos alunos participantes das apresentações e aulas síncronas ao vivo, não sendo permitida a utilização das imagens, áudio ou conteúdo dos vídeos para outros fins que não os objetivos pedagógicos explicitamente definidos para esta disciplina. O uso indevido da imagem de professor e dos alunos é crime previsto na Constituição Federal. Por este motivo não é permitido aos alunos gravar e compartilhar imagens e falas de docentes e discentes. A participação do aluno na aula síncrona ao vivo implica na sua autorização implícita para a gravação da aula. Se por acaso algum aluno não se sentir confortável em participar da aula síncrona ao vivo por estar sendo gravada, poderá não participar ao vivo e assistir a aula gravada posteriormente.

Conforme parágrafo 2º do artigo 70 da Resolução 17/CUn/97, o aluno com frequência suficiente (FS) e média final no período (MF) entre 3,0 e 5,5 terá direito a uma nova avaliação ao final do semestre (REC), sendo a nota final (NF) calculada conforme parágrafo 3º do artigo 71 desta resolução, ou seja: $NF = (MF + REC) / 2$.

10) Cronograma

- Semana 1 - Apresentação da disciplina. Introdução à programação C++ e ao ambiente de desenvolvimento
- Semana 2 - Depuração de programas em ambientes de programação, criação de testes unitários, gerência e alocação dinâmica de memória
- Semana 3 - Modelagem e programação de Pilha e Fila em vetor (arrays)
- Semana 4 - Lista em vetor (array) como caso geral de Pilha e Fila
- Semana 5 - Lista Encadeada
- Semana 6 - Fila Encadeada e Pilha Encadeada como especializações de Listas Encadeadas.
- Semana 7 - Lista Circular. Enunciado do Projeto I
- Semana 8 - Conceito de Complexidade de Algoritmos
- Semana 9 - Prova Teórica I (no Moodle) e Prova Prática I (no Moodle)
- Semana 10 - Árvores Binárias de Busca. Árvores Binárias de Busca Semibalanceadas.
- Semana 11 - Auxílio com trabalhos pendentes. Apresentação do Projeto I
- Semana 12 - Árvore AVL. Enunciado do Projeto II
- Semana 13 - Árvore Rubro-Negra (Red-Black)
- Semana 14 - Gerência de Arquivos. Árvores de Busca Semibalanceadas Multivias
- Semana 15 - Hashing. Lista invertida
- Semana 16 - Métodos de Ordenação - Quicksort e Heapsort
- Semana 17 - Prova Prática II (no Moodle) e Prova Teórica II (no Moodle)
- Semana 18 - Auxílio com trabalhos pendentes. Apresentação do Projeto II

11) Bibliografia Básica

- DROZDEK, A. Estrutura de Dados e Algoritmos em C++. 4. ed. Cengage Learning, 2017. ISBN: 9788522126651. Disponível em: <https://www.cengage.com.br/ls/ebook-estrutura-de-dados-e-algoritmos-em-c/> (acesso via <http://portal.bu.ufsc.br/>)
- MORIN, P. Open Data Structures (in C++). Edition 0.1Gβ. Disponível em: <http://opendatastructures.org/>
- LUCCHESI, C.L., KOWALTOWSKI, T. Estruturas de Dados e Técnicas de Programação. Instituto de Computação, UNICAMP, 2004. Disponível em: https://www.ic.unicamp.br/~mc202abcd/ln_tkcll.pdf
- RICARTE, I.L.M. Estruturas de dados. Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 2008. Disponível em: <http://calhau.dca.fee.unicamp.br/wiki/images/0/01/EstruturasDados.pdf>

12) Bibliografia Complementar

- Materiais disponibilizados via Moodle.
- PEREIRA, Silvio do Lago. Estruturas de dados fundamentais: conceitos e aplicações. 12. ed., rev. e atual. São Paulo: Érica, 2009. 238 p. ISBN 9788571943704
- JOYANES AGUILAR, Luis. Programação em C++: algoritmos, estruturas de dados e objetos. São Paulo: McGraw Hill, 2008. xxxi, 768 p. ISBN 9788586804816
- WIRTH, Niklaus; ZÜRICH, Eth. Algoritmos e estruturas de dados. Rio de Janeiro: LTC, 1999. 255 p. ISBN 978-85-216-1190-5
- HOROWITZ, Ellis. Fundamentos de estruturas de dados.. Rio de Janeiro: Ed. Campus, 1986.
- CLAYBROOK, Billy G. (Billy Gene). Técnicas de gerenciamento de arquivos. 2. ed. Rio de Janeiro: Campus, 1987. 248p. ISBN 8570012608
- SILBERSCHATZ, KORTH E SUDARSHAN. Database System Concepts. 4ª ed., McGraw-Hill, 2001.
- THARP, A. File Organization and Processing. John Wiley, 1988.
- SEDGEWICK, R. Algorithms in C++. Addison Wesley, 1996.
- GOODRICH, M.T., TAMASSIA, R. Estruturas de Dados e Algoritmos em Java. 4a. Edição. Ed. Bookman, 2007.
- AMERAAL, L. Algorithms and Data Structures in C++. Editora John Wiley, 1996.
- TENNENBAUM, A., LANGSAM, Y. Data Structures using C and C++. 2ª Ed. Prentice-Hall, 1995.